

IN THE CLAIMS

Please amend the claims as follows.

1. (Withdrawn) A method to manage interactions between applications and a data store, comprising:

- (a) concurrently extracting data from a data store into a first queue;
- (b) concurrently servicing a plurality of applications with portions of the data from the first queue;
- (c) concurrently loading results data into a second queue, wherein the plurality of applications concurrently produce the results data;
- (d) concurrently populating the results data into a temporary table; and
- (e) merging the temporary table with a data store table of the data store.

2. (Withdrawn) The method of claim 1 further comprising:

- (f) establishing a plurality of first queues, wherein each first queue is associated with a separate processing node that executes a subset of the plurality of applications and each first queue includes the extracted data and concurrently services the subset of the plurality of applications on the separate processing node; and
- (g) establishing a plurality of second queues, wherein each second queue is associated with the separate processing node that executes the subset of the plurality of applications and each second queue is currently loaded with results data from the subset of the plurality of applications and concurrently populates a second table, which is merged with the temporary table before performing the merging.

3. (Withdrawn) The method of claim 1 wherein (b) further includes concurrently servicing a number of the applications from a first processing node and concurrently servicing a remainder of the applications from a second processing node, wherein the first queue resides on the first processing node.

4. (Withdrawn) The method of claim 1 wherein (c) further includes currently loading the second queue with portions of the results data acquired from a number of the applications processing on a first processing node and remaining portions of the results data acquired from a remainder of the applications processing on a second processing node, wherein the second queue resides on the first processing node.

5. (Withdrawn) The method of claim 1 wherein (d) further includes currently populating the temporary table with portions of the results data received from the second queue associated with a number of the applications processing on a first processing node and with remaining portions of the results data received from a different second queue associated with a remainder of the applications processing on a second processing node, wherein the second queue resides on the first processing node and the different second queue resides on the second processing node.

6. (Withdrawn) The method of claim 1 further comprising (f) concurrently initiating a number of the applications on a first processing node and a remainder of the applications on a second processing node.

7. (Withdrawn) The method of claim 6 processing further comprising concurrently establishing the processing of (b)-(d) on the first processing node and the second processing node.

8. (Currently Amended) A method to manage interactions between applications and a data store, comprising:

receiving a query for a data store and an identifier for an application, wherein the application when executed seeks to process results returned from and produced by executing the query and seeks to update the data store with application data, wherein the application data is produced in response to the application processing the results of the query;

concurrently executing multiple instances of an application associated with the identifier on multiple processing nodes within a network to achieve parallel processing for the multiple instances of the application;

concurrently processing the query to acquire the results on behalf of the multiple instances of the application and producing the results that are then housed in one or more application queues residing on one or more of the processing nodes, each application queue having different portions of the results; and

concurrently providing the results to each of the instances of the application from the one or more application queues so that the instances can produce the application data from the results and update the data store with the application data, which is to be subsequently accessed from the data store.

9. (Original) The method of claim 8 further comprising:

concurrently housing the application data in one or more load queues residing on one or more of the processing nodes; and

concurrently populating one or more tables residing on one or more of the processing nodes with the application data from the one or more load queues.

10. (Original) The method of claim 9 further comprising merging the one or more tables into the data store.

11. (Original) The method of claim 8 wherein the currently initiating further includes determining a total number of the applications to initiate based on configuration data.

12. (Original) The method of claim 11 wherein the currently initiating further includes determining which of a number of the applications that are to be initiated on which of a number of the processing nodes based on the configuration data.

13. (Original) The method of claim 8 further comprising concurrently synchronizing the application queues and the load queues on the multiple processing nodes when at least some of the processing nodes lack one of the one or more application queues or one of the one or more load queues.

14. (Original) The solution template system of claim 13 wherein the concurrently synchronizing further includes establishing socket based communications between the multiple processing nodes with a Transmission Control Protocol/Internet Protocol (TCP/IP).

15. (Currently Amended) A data store application management system implemented in a processing and memory device, comprising:

one or more application queues for servicing one or more applications with results of a query to a data store, wherein the one or more application queues reside within memory of the device and receive the results when the query executes, and wherein each application queue includes a different portion of the results;

one or more load queues for housing application data produced by the one or more applications that process the results from the one or more application queues, wherein the one or more load queues reside within memory of the device and receive the application data when the applications execute in parallel with one another as application instances on multiple node devices of a network; and

a merge utility for merging the application data into a data store table, wherein the merge utility executes on the device and merges the one or more load queues into the data store table to update the data store with the application data that is to be subsequently accessed from the data store.

16. (Original) The system of claim 15 further comprising a configuring utility for determining a total number of the one or more applications.
17. (Original) The system of claim 15, wherein the configuring utility initiates a number of the one or more applications, the one or more application queues, and the one or more load queues on separate processing nodes.
18. (Original) The system of claim 15, wherein each of the one or more applications concurrently processes the results and produces different portions of the application data.
19. (Original) The system of claim 18, wherein each of the one or more application queues and each of the one or more load queues concurrently update while the one or more applications process.
20. (Currently Amended) A data store implemented in a processing and memory device and accessible over a network, comprising:
- one or more temporary tables that temporarily house application data produced from concurrently processing applications in response to concurrently provided query results extracted from the data store, wherein a query is executed in parallel over the network on multiple nodes and the query results are provided to the applications that are processing in parallel over the network on the multiple nodes and are instances of one another, the applications consume the query results to produce the application data, which is then housed in the one or more temporary tables, and wherein when the query is executed the query results are extracted and different portions of the results are provided to each of the applications for parallel processing of the results with one another and for producing the application data, each application producing a different portion of the application data; and
- an application data table that houses the application data once the applications have finished producing the application data, and wherein the one or more temporary tables are merged into the application data table and populate the application data table with the application

data, the application data represents processing results of the applications produced in response to the query results and the application data is housed in the application data table of the data store for subsequent access.

21. (Original) The data store of claim 20 wherein a merge utility merges the one or more temporary tables to produce the application data table once each of the plurality of applications have finished processing the query results.
22. (Original) The data store of claim 20 wherein one or more extract utilities perform a query against the data store in order to acquire the query results, which are concurrently consumed by the one or more applications to produce the application data.
23. (Original) The data store of claim 22 wherein each of the one or more extract utilities concurrently populate the query results to one or more application queues.
24. (Original) The data store of claim 23 wherein each of one or more load utilities concurrently receive portions of the application data from one or more load queues and concurrently populate the portions to the one or more temporary tables.
25. (Original) The data store of claim 22 wherein the data store is a least one of one or more databases and a data warehouse.